(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 May 2001 (10.05.2001)

PCT

(10) International Publication Number
WO 01/33456 A2

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US00/29571

(22) International Filing Date: 27 October 2000 (27.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/162,485 29 October 1999 (29.10.1999) US

(71) Applicant: AP ENGINES, INC. [US/US]; One Clock Tower Place, Maynard, MA 01754 (US).

(72) Inventor: WILCOX, William; 9 Summit Street, Maynard, MA 01754 (US).

(74) Agent: DINIUS, Anne-Marie; Choate, Hall & Stewart, 53 State Street, Exchange Place, Boston, MA 02109 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/33456 A2

(54) Title: SYSTEM AND METHOD FOR BILLING NETWORK USERS

(57) Abstract: The present invention provides a system and method for billing Internet users based upon the amount of bandwidth they consume when they access the Internet. The method comprises a software package referred to as the billing rule. A service provider using the billing rule is able to enter parameters into a separate data file, which results in a custom tailored billing scheme. Similarly, the system disclosed herein facilitates Internet access in usage-based fashion and includes an Internet Protocol network, a hardware operating environment, an interface, and the billing rule software.

System and Method for Billing Network Users

Background of the Invention

This application claims the benefit of priority of U.S. Patent Application Serial No. 60/162,485, filed October 29, 1999, entitled "Billing Rule," the teachings of which are incorporated herein by reference.

The Internet is a cooperative message-forwarding system linking computer networks all over the world. Users of the Internet can exchange electronic mail, participate in electronic discussion forums, send files from any computer to any other computer, retrieve information, and even use each others computers directly if they have appropriate passwords. DOWNING ET AL, DICTIONARY OF COMPUTER AND INTERNET TERMS, 239-40 (6th ed. 1998).

Individual users seeking to use the Internet typically gain access via online service providers. The cost of running the Internet is paid primarily by these service providers who, in turn, attempt to fairly distribute those costs among their customers. In today's market, service providers charge a flat rate for providing Internet access. This flat rate is either a true flat rate, e.g., a fixed price per month for unlimited access, or it is a flat hourly rate assessed to customers based upon the actual time that they are accessing the Internet.

Since neither of these billing schemes depends upon the actual bandwidth consumed by individual users, the allocation of costs to customers is inefficient and unfair. The following example illustrates this inequity. Two residential users can often times consume vastly different amounts of bandwidth during the time that they access the Internet. A power user may, for example, be downloading graphical or video data, listening to MP3 files, and operating a personal web page. A less savvy user, on the other hand, may use his or her Internet account to send email and to occasionally surf the web. If the service provider for these two users charges them a fixed monthly rate, they will incur the same cost despite having used vastly different system resources. Similarly, if the service provider charges an hourly rate for Internet access and these two users access the Internet for the same amount of time in a given billing cycle, their bills will again be identical even though they consumed vastly different system resources while they were accessing the Internet.

In addition to unfairly allocating costs by failing to account for bandwidth consumption, the present billing schemes used by service providers also overlook fluctuations in system performance. During peak hours, data packets are transported through the various routers, switches, and hubs that form the backbone of the Internet at a much slower rate than during off-peak hours. This fluctuation in system performance is not factored into the standard flat-rate billing schemes presently employed by service providers. The present invention overcomes these drawbacks by providing service providers with a more flexible billing framework, which accounts for actual bandwidth consumption.

Summary of the Invention

In one aspect, the invention comprises a method for billing users for Internet access based upon the actual bandwidth consumed by the user while he or she is accessing the Internet. The method can be practiced by executing a software package referred to herein as the billing rule. According to this aspect of the invention, a service provider is capable of specifying billing criteria, which is then used by the billing rule to determine if and how much a user should be billed for accessing the Internet. The service provider could, for example, bill users based upon the time of day that they are accessing the Internet, charging higher rates for peak-use periods. In addition, service providers can monitor the total erroneous data bytes transmitted to or from a network user's device. In this way, the service provider could reduce a user's bill if he or she encountered an unusually high amount of erroneous data transmissions during a particular billing cycle.

In another aspect, the invention comprises a system that implements the billing rule software. This aspect of the invention is comprised of an Internet Protocol network, a hardware operating platform, the billing rule software, and an interface between the billing rule software and the hardware operating platform. The system is capable of communicating with network user's devices in order to determine their bandwidth consumption. This communication is facilitated by executing the billing rule software. The billing rule software can be stored in the internal memory of the hardware operating platform.

Brief Description of the Drawing

The invention is described with reference to the several figures of the drawing, in which,

Figure 1 is a flow chart illustrating a method for billing a network user for accessing the Internet;

Figure 2 is a flow chart showing the creation of a device object;

Figure 3 is a flow chart depicting the functions performed by the class of methods comprising the billing rule; and

Figure 4 is a block diagram showing a system for billing network users for Internet access based upon bandwidth consumption.

Detailed Description

The invention disclosed herein provides an alternative method and system by which service providers can bill network users for providing them with access to a network. The types of service providers who may find this invention useful include those providing network access via telephone lines, cable, or through the airwaves. In one aspect of the invention, a method that allows service providers to bill network users for accessing the Internet is discussed. Unlike many prior art billing schemes which bill network users at a flat rate, network users in a preferred embodiment of the present invention are billed according to their actual bandwidth consumption. The billing rule disclosed herein may, for example, be a Java class of methods that can be run by a service provider.

In a preferred embodiment of the billing rule, a network user is billed for accessing the Internet according to the steps shown in **Figure 1**. According to this embodiment, a service provider must first create **10** a file of type .Java. The .Java file is then compiled **20** into a file of type .class with a Java compiler. The .class file is then loaded into a polling engine where it will eventually be used by the polling engine to determine what criteria should be used when billing network users. According to this embodiment, the service provider could use a Microsoft Windows-type management interface when creating **10** the Java class file containing operational parameters. The operational parameters specify which data counters the service provider would like the polling engine to base its billing calculations on during

execution of the billing rule. The operational parameters can be chosen from the following list of data counters:

- ifInOctets - number of bytes received
- ifInUcastPkts - number of unicast packets received
- ifInNUcastPkts - number of non-unicast packets received
- ifInDiscards - number of inbound packets discarded
- ifInErrors - number of packets received in error
- ifInUnknownProtos - number of packets received with unknown protocol type
- ifOutOctets - number of bytes sent
- ifOutUcastPkts - number of unicast packets sent
- ifOutNUcastPkts - number of non-unicast packets sent
- ifOutDiscards - number of outbound packets discarded
- ifOutErrors - number of errors on outbound packets

The data counters are described more fully in the Internet Engineering Task Force's Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II, RFC 1213, the contents of which are incorporated herein by reference. As can be seen from the above listing, each data counter represents an aspect of the bandwidth consumption of a network user's device. In accordance with this invention, the data counters are retrieved by the polling engine. The polling engine accumulates the results of polling and passes these results to the billing rule at the end of the polling interval.

When the service provider creates **10** the Java class file, he or she chooses which of these data counters will be collected and stored by the polling engine. These choices become the operational parameters and are stored in a Java class file. In an additional embodiment of the present invention, the service provider could alter the cost of accessing the Internet based upon such factors as time of day that the network user accessed the Internet, whether the total bytes of data transmitted and received during a particular session exceeded a threshold value, or whether the entity accessing the network is an individual or a business. If the service provide chose to structure billing in this way, it would include these criteria in the Java class file.

In a preferred embodiment, a polling engine executes the billing rule software. The billing rule software could be a Java class of methods. As part of its duties in executing the billing rule, the polling engine polls each network device that is accessing the Internet. In these polling exchanges, the polling engine accumulates data for each of the operational parameters specified by the service provider. Thus, in a preferred embodiment, a service provider can create a Java class file directing the polling engine to accumulate data for as many of the counters listed above as the service provider wishes to monitor. In addition, service providers can create many different Java class files, enabling the polling engine to monitor different counter data for different network users. After the service provider compiles **20** the Java class file, it is stored **30** in a data storage unit. The data storage unit could be a database, computer hard drive, or the like.

Once the Java class file has been stored **30**, some of the operational parameters, comprising such information as an IP address and a billing rule name associated with a particular network user, are provided **40** to the software via a web browser or a text file. The polling engine uses this IP address and billing rule name to create **50** the device object. Each device object created **50** by the polling engine represents a unique device configured to operate according to this aspect of the invention. Since it is likely that a plurality of devices will be accessing the network according to the invention disclosed herein, it may be necessary for the polling engine to create a plurality of device objects, each of which is used to distinguish one device from the next. Once a first device object corresponding to a first network device has been created, the first device object uses a first billing rule name passed from the web browser or text file to load a Java class file containing the billing rule byte code. At this point, additional operational parameters, such as the polling interval and the polling period each discussed further below, are extracted from the billing rule. After this step is complete, the device object has been constructed and may comprise: (1) a first device's IP address; (2) a first Simple Network Management Protocol ("SNMP") port number; (3) a first billing rule name; and (4) a first billing rule parameter.

Figure 2 shows a detailed depiction of how the polling engine creates **50** the device object. As a first step in this, the first device object dynamically loads **51** the first billing rule into a Java virtual machine. After the first billing rule has been

loaded, the first device object creates **52** an instance of the first billing rule. At this point, the Java runtime constructor invokes **53** a first billing rule default constructor. As can be seen from **Figure 2**, the next step entails the initialize method of the first billing rule being called. When the initialize method of the first billing rule is called, a billing rule parameter is passed **54** as the sole parameter. The first device object then allocates **55** the accumulator array, which is where the polling engine will eventually store the results it obtains. The final step in this embodiment comprises the first device object creating **56** the necessary structures for the SNMP polling of the first device. In addition, the first device object maintains a copy of the operational parameter indicating the start of the polling interval. Once the first device object has initialized the first billing rule, the polling engine places the first device object on a polling list. In a preferred embodiment, the polling engine continuously loops through the polling list, sequentially polling each of a plurality of devices.

After the first device object has been created, it is possible for the polling engine to call the remaining methods in the billing rule. These methods, and the order in which they are called, are depicted in **Figure 3**. As can be seen from **Figure 3**, the first step performed during execution of the billing rule is an initialization **110**. In this embodiment, the first device object is used to initialize a first billing rule. After initialization has been completed for the first device, the polling engine determines **120** the polling interval and the polling period from the operational parameters in the first Java class file. The begin time for the polling interval is a Gregorian Calendar object, which is part of the standard Java Application Programming Interface. The frequency at which polling will transpire during the polling interval is referred to as the polling period and may be given in seconds. The counters that can be polled are chosen from the list set forth above with reference to the MIB-II, RFC 1213 standard. At the end of the polling interval, the results that were accumulated are delivered to the polling engine.

After these preliminary operational parameters have been extracted from the first Java class file, the polling engine begins polling the first device and accumulating **130** values for the counters specified by the service provider. During each poll, the polling engine determines **140** if the polling interval has ended by comparing the current time to the end of the polling interval as specified by the billing rule. The

method used to determine if polling interval has ended is also a Gregorian Calendar object. Once the polling interval has ended, the polling engine stops accumulating counter data, calls a method in the billing rule for generating usage billing data, passes the accumulator array, and sends **150** a data array of the specified counters to the billing rule so that the billing rule can compute the usage billing data for the first device. The data array contains the accumulated counters gathered during the polling interval.

When the billing rule finishes computing the usage billing for the first device, it passes one of two values back to the polling engine. The billing rule returns either a null value, indicating a non-billable event, or it returns data constituting a billable event. If the value returned is null, the polling engine resets **180** the billing rule. If, on the other hand, a billable event is returned, the polling engine may instantiate and pass **170** a billable event object downstream. The billable event object could be passed downstream to another component via an interface. In one embodiment, this interface could be a CORBA device. In another embodiment, the polling engine may simply pass the billable event object downstream across a network to a computer component, which in turn determines the appropriate action to take with respect to the billable event object. An example of an appropriate action might be to reformat the event and pass it on to a billing system maintained by the service provider, thereby placing a charge on the network user's account.

Distinguishing a billable event from a non-billable event can depend on a number of factors determined ahead of time by the service provider when it establishes the operational parameters in the Java class file. The operational parameters described here are actually an arbitrarily complex piece of Java code. For example, the service provider may stipulate that it will not generate a billable event for Internet access if the network user consumed less than a threshold value of bandwidth while accessing the Internet. Alternatively, the service provider could determine that it will not bill network users for Internet access of less than a threshold value of minutes, or for use occurring during a particularly low usage period. Because the polling interval and the polling period are retrieved from the billing rule at runtime, the values returned by the billing rule from one polling interval to the next

are computed using java code, providing the operator with maximum flexibility in determining their respective values.

Referring again to **Figure 3**, resetting **180** the polling engine clears the counter arrays and prepares the polling engine for the next polling interval. After the polling engine has been reset, it determines **120** when the next polling interval should begin and the polling period, which is the frequency at which the network user's device will be polled.

In an additional embodiment, the polling engine can maintain accumulated values for the various data counters polled on each network device as 64-bit values. Moreover, the polling engine can subtract an internally computed amount of overhead associated with polling each device so that network users are not charged for being connected to the network during the time that the polling engine is polling a particular device.

In a preferred embodiment shown in **Figure 4**, a system for billing network users could be implemented on an Internet Protocol, commonly known as an IP, network **310**. The IP network in a preferred embodiment may further comprise a web browser. One function of the web browser could be to allow input of the parameters for polling a single network user's device and to otherwise manage the system.

The hardware operating platform **320** in a preferred embodiment may be comprised of a computer, a processor **330**, and interface **335**, two Local Area Network cards, **340** and **345**, respectively, and a data storage unit **350**. Once the service provider configures the hardware operating platform **320**, he or she may load the billing rule software into the running system from a predefined location in the installation tree. In this way, the hardware operating platform **320** can be used to store the billing rule software. In a preferred embodiment, the service provider can remotely access the billing rule software via the Internet. Additionally, a text file comprising a listing of the plurality of network user's devices **360**, which are configured to operate on a usage basis, can also be stored in the data storage unit **350**.

Other embodiments of the invention will be apparent to those skilled in the art from a consideration of the specification or practice of the invention disclosed herein.

It is intended that the specification and examples be considered as exemplary only, with the true scope and spirit of the invention being indicated by the following claims.

What is claimed is:

1. A computer implemented method for billing a network user for access to the Internet, comprising the steps of:

creating a data file containing operational parameters;

storing the data file in a data storage unit;

executing a computer readable program, wherein the computer readable program extracts an operational parameter from the data file, polls a network user's device, receives data values from the network user's device in response to the polling, processes the data received during the polling, and determines whether the processed polling data constitutes a billable event.

2. The computer implemented method of claim 1, wherein the data file is compiled to be a Java class file.

3. The computer implemented method of claim 1, wherein the operational parameters comprise counter data for data transmitted or received by the network device, the counter data selected from the group consisting of bytes of data received, unicast packets received, non-unicast packets received, inbound packets discarded, packets received in error, packets received with unknown protocol types, bytes sent, unicast packets sent, non-unicast packets sent, outbound packets discarded, and errors on outbound packets.

4. The computer implemented method of claim 1, further comprising transmitting the billable event to a service provider.

5. A computer implemented method comprising the steps of:

initializing a billing rule;

placing a device object on a polling list;

extracting operational parameters from a data file;

polling a network user's device;

accumulating an array of counter data;

determining whether to continue polling;

processing the array of counter data;

transmitting the processed counter data to the billing rule; and

resetting a polling engine.

6. The method of claim 5 wherein the data file is a Java class file.

7. A system for billing network users, comprising:

a network, wherein the network is an Internet Protocol network;

a hardware operating platform;

an interface; and

a computer readable billing rule, wherein the computer readable billing rule further comprising a class of methods.

8. The system of claim 7, wherein the Internet Protocol network has a web browser.

9. The system of claim 7, wherein the hardware operating system further comprises a computer, a processor, a local area network card, and a data storage unit.

10. The system of claim 7, wherein the interface is a Java virtual machine.

FIG. 1

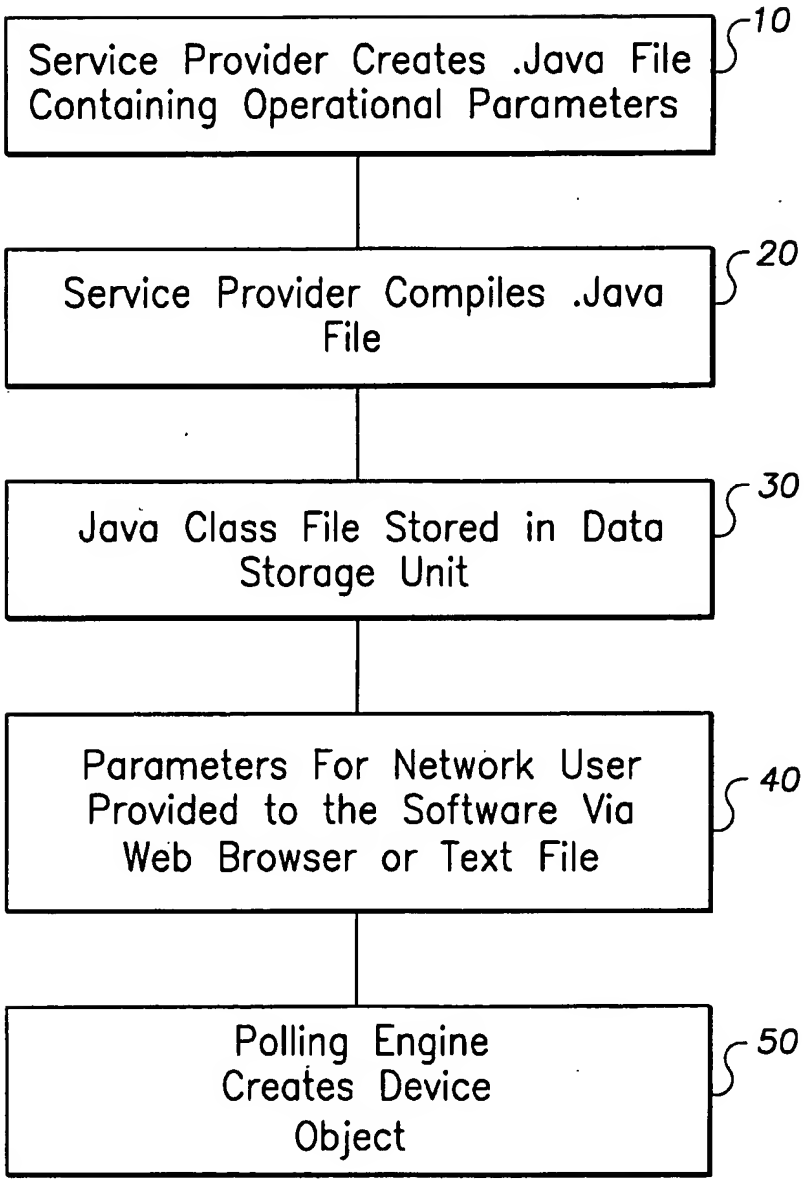

Service Provider Creates .Java File Containing Operational Parameters
10
Service Provider Compiles .Java File
20
Java Class File Stored in Data Storage Unit
30
Parameters For Network User Provided to the Software Via Web Browser or Text File
40
Polling Engine Creates Device Object
50

FIG.2

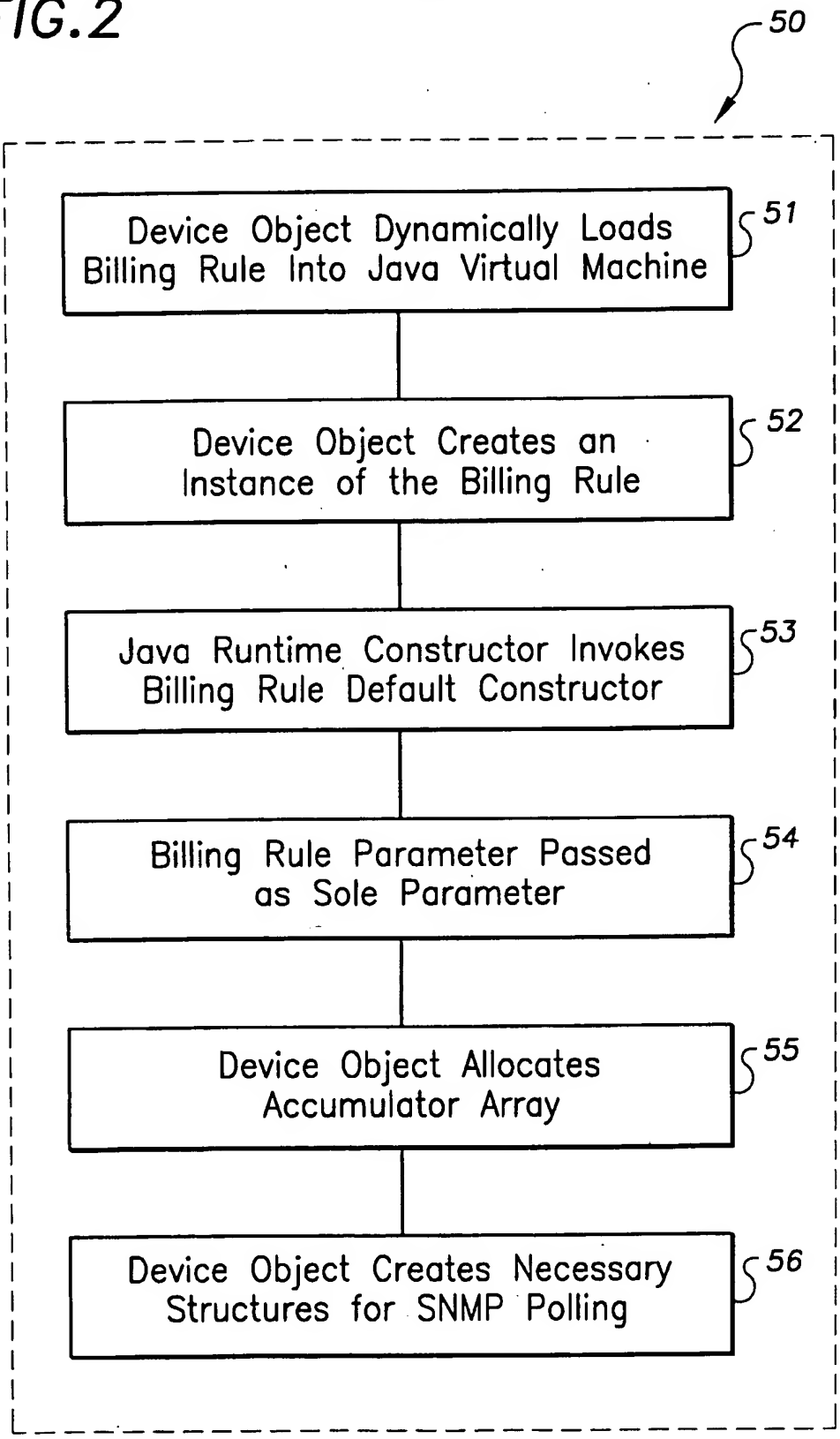
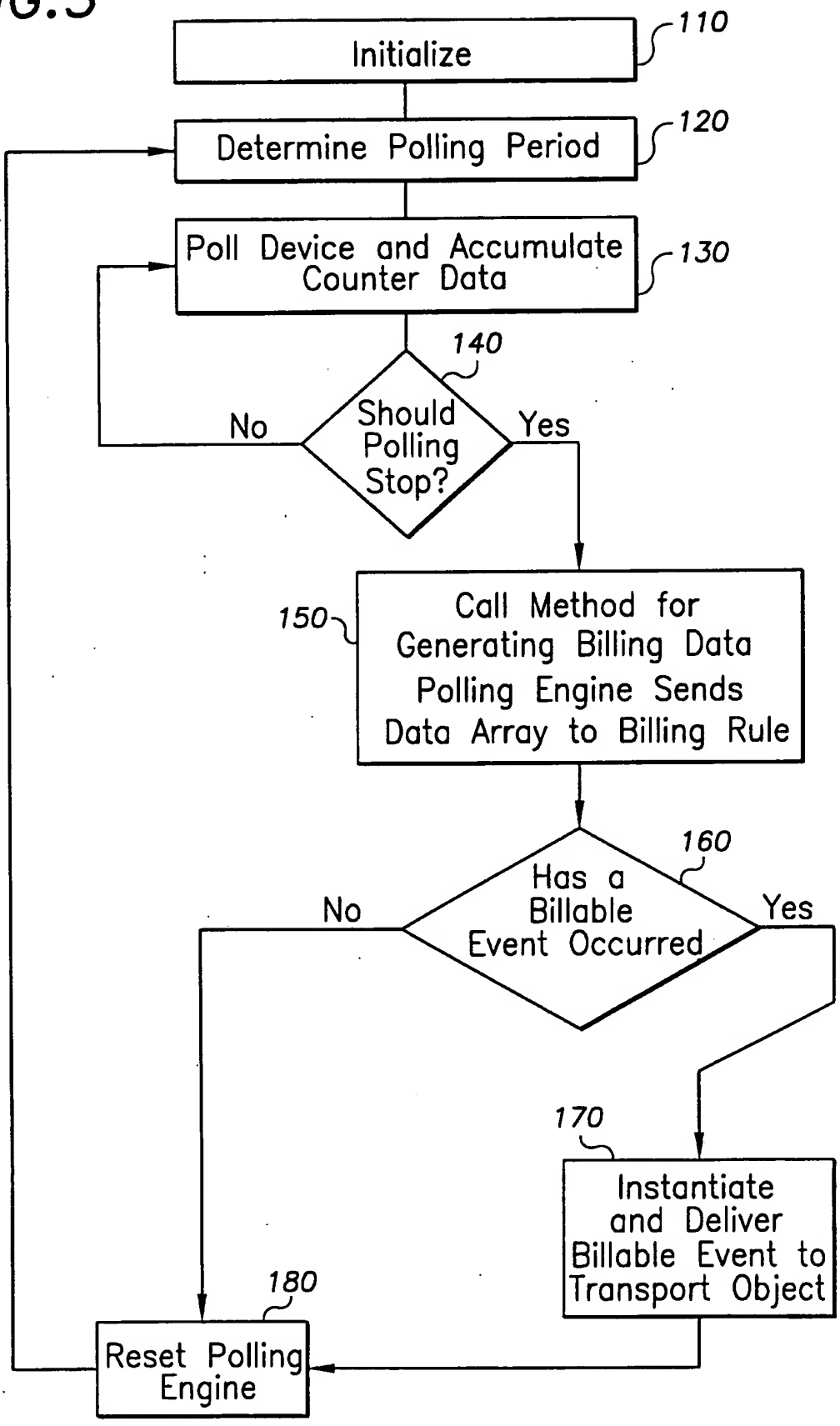

50
Device Object Dynamically Loads Billing Rule Into Java Virtual Machine
51
Device Object Creates an Instance of the Billing Rule
52
Java Runtime Constructor Invokes Billing Rule Default Constructor
53
Billing Rule Parameter Passed as Sole Parameter
54
Device Object Allocates Accumulator Array
55
Device Object Creates Necessary Structures for SNMP Polling
56

FIG.3


Initialize
110
Determine Polling Period
120
Poll Device and Accumulate Counter Data
130
140
Should Polling Stop?
No
Yes
150
Call Method for Generating Billing Data Polling Engine Sends Data Array to Billing Rule
160
Has a Billable Event Occurred
No
Yes
170
Instantiate and Deliver Billable Event to Transport Object
180
Reset Polling Engine

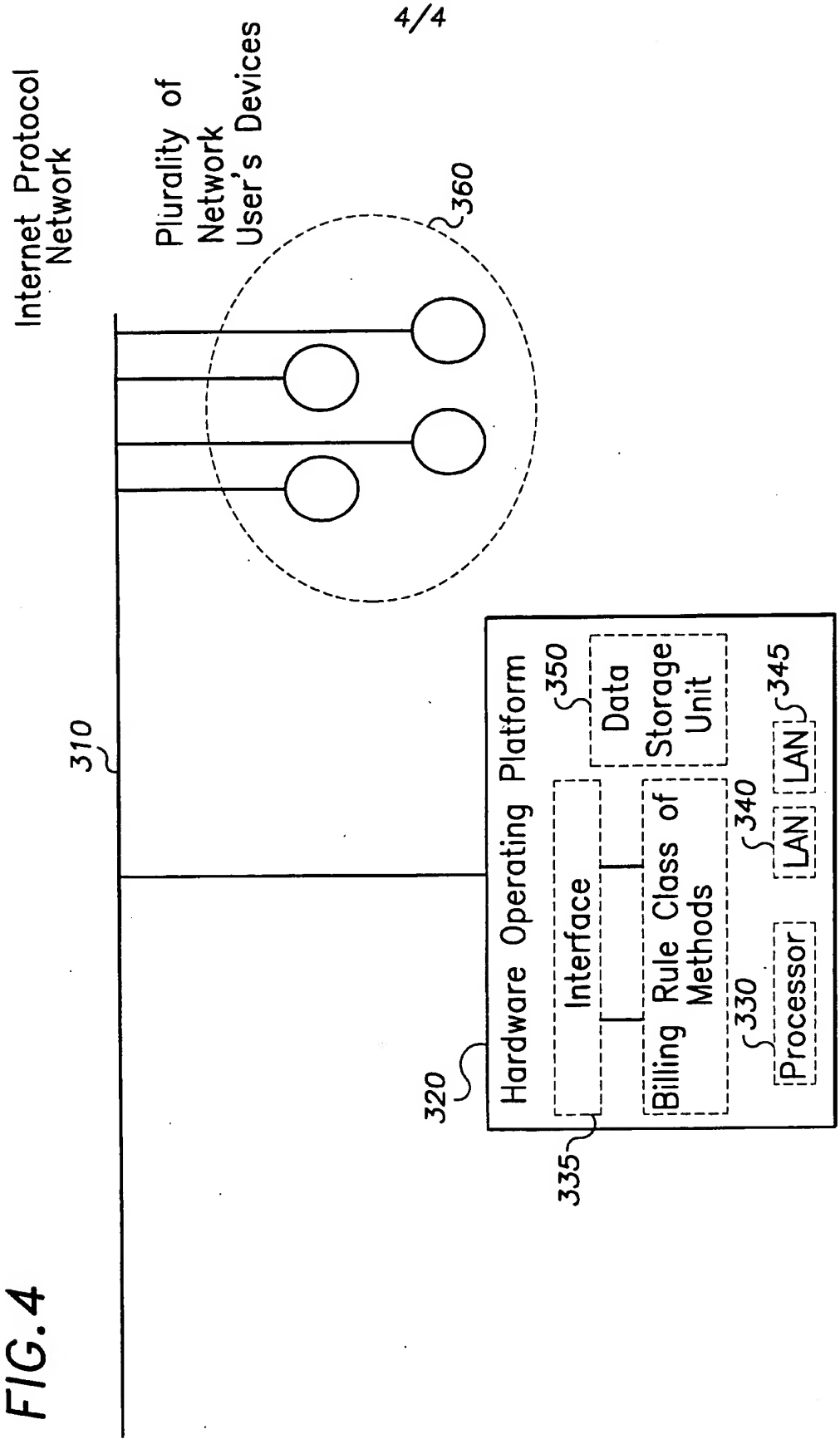

Internet Protocol Network
Plurality of Network User's Devices
360
310
Hardware Operating Platform
320
Interface
335
Billing Rule Class of Methods
330
Processor
340
LAN
LAN
345
Data Storage Unit
350

FIG. 4